OIPE
DEC 06 2004
PATENT & TRADEMARK OFFICE

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Oleg Kiselev and John A. Colgrove		
Assignee:	VERITAS Software Corporation		
Title:	AUTOMATED RECOVERY FROM DATA CORRUPTION OF DATA VOLUMES IN RAID STORAGE		
Application No.:	10/610,966	Filed:	July 1, 2003
Examiner:	Brian L. Johnson Special Programs Examiner Technology Center 2100 703-308-0885	Group Art Unit:	2186
Docket No.:	VRT0061US	Confirmation No.:	1460

Austin, Texas
December 3, 2004

Mail Stop PETITION
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

RENWEWED PETITION TO MAKE SPECIAL UNDER 37 CFR §1.102(d)

Dear Sir:

Applicants acknowledge receipt of the September 8, 2004 decision dismissing their initial petition to make special. The initial petition was dismissed for failure to adequately meet the requirement (e) of MPEP § 708.02. More particularly, the decision indicates that a complete detailed discussion of the most closely related references has not been provided with the necessary specificity required under 37 C.F.R. 1.111 (b) and (c).

Applicants submit this renewed petition to make special. In that the fee of $130.00 for a petition to make special was previously paid along with the initial petition to make special, Applicants submit that no further fee is required. However, if a further

12/07/2004 JBALINAN 00000015 502306 10610966
02 FC:1464 130.00 DA

fee is required, Applicants authorize Patent Office to charge Deposit Account No. 502306.

Should the Office determine that all the claims presented are not obviously directed to a single invention, the applicants will make an election without traverse as a prerequisite to the grant of special status.

The applicants respectfully submit that a pre-examination search has been performed by a professional search firm in the following classes/subclasses:

Class	Subclasses
G06F (Electrical Digital Data Processing)	03/06 13/28 11/10 11/00 11/20 11/14 12/08 12/28 17/30 12/08
714 (Error Detection/Correction and Fault Detection/Recovery)	006 041 766 801 805
707 (Digital Processing: Database and File Management or Data Structures)	100

Enclosed are copies of the following references which are presently believed to be, from among those made of record in the accompanying Information Disclosure Statement and any previously filed Information Disclosure Statement, the most closely related to the subject matter encompassed by the claims:

U.S. Patent Publication US 2002/0169995 A1	John E. Archibald Jr. et al.
U.S. Patent No. 6,606,629 B1	Rodney A. DeKoning et al.
U.S. Patent No. 5,948,110	David Hitz et al.
U.S. Patent No. 5,623,595	William Bailey
European Application EP 0 837 393 A2	David Gordon et al.

Detailed Discussion of the References

U.S. Patent Application Publication 2002/0169995 A1 of Archibald, Jr. et al. ("Archibald") was filed May 10, 2001 and published November 14, 2002. Archibald relates to a system, method, and computer program for selectable or programmable data consistency checking methodology. Figure 1 of Archibald illustrates a block diagram of a redundant array of independent disks (RAID) data storage system 100. Figure 2 is a block diagram that illustrates aspects of an exemplary data stripe according to Archibald.

Archibald describes in paragraph 0005 that prior art data consistency checking techniques are limited in that they typically have no means or only limited means for determining whether the user data is corrupt or whether the parity data is corrupt when an inconsistency is identified. This limitation may result in erroneous data being stored in a data stripe when a data consistency check with an auto-correction is utilized. To illustrate such an erroneous result, consider that in some prior art techniques, upon detecting such an inconsistency, only the corresponding parity data is altered to make it consistent with the user data. In such techniques, having no other information, an assumption is made that the parity data has become corrupted rather than the user data. As a result, if the user data was actually corrupt, but not the parity data, the user data typically remains corrupt, and parity data consistent with the corrupt user data is written to the data stripe. Archibald that states that what is needed is a system, method and computer program that can differentiate between corrupted parity data and corrupted user data, such that corrupted parity data is appropriately corrected in view of uncorrupted user data, and such that corrupted user data is corrected in view of uncorrupted parity data.

The summary of Archibald sets forth a first procedure that provides a method to perform data consistency checking and correction of user data stored in data sectors on a data stripe. The summary states:

> Each data sector includes data header and data stripe includes a parity sector, and parity sector includes a parity header. Data stripe

> is distributed across storage devices. Provides each data header with first code to provide a check for the user data stored in the data header's data segment, provides the parity header with second code to provide a check for a set of parity data stored in parity sector, and provides the parity header with third code by encoding each respective first code with second code. Third code enables user to check consistency of user data and parity data without respective operations on each byte of data in both data sectors and parity sector.

With reference to Figures 1 and 2, Figure 3 is a flow chart that illustrates aspects of a procedure to perform data consistency checking according to the invention of Archibald. The process starts in block 302 when a data stripe is read. In block 303, 308 a first or next sector stripe is selected. Thereafter in block 304, a data check code sub-sector stripe (DCC_{ss}) is calculated. The calculated DCC_{ss} is compared to a stored DCC_{ss} for the selected sector stripe. If the calculated DCC_{ss} is equal to the stored DCC_{ss}, then the process repeats with block 303, 308 if there are more sector stripes. If the calculated DCC_{ss} does not equal the stored DCC_{ss} for the selected sector strip, then either the stored DCC_{ss} is replaced with the calculated DCC_{ss} or an error is determined and a determination is made as to whether the error is correctable.

Paragraph 33 indicates that, with reference to Figure 2, parity header 202 includes a DCC_{ss} value derived from DCC_{ps} value in the parity header 202 and the respective DCC_{ds} values associated with the sectors. In one embodiment, the DCC_{ss} is a summation of the DCC_{ps} and the DCC_{ds} values in the sector stripe. In the preferred embodiment of Archibald, the DCC_{ss} value is an XOR of the DCC_{ps} value and the DCC_{ds} values in the sector stripe. Paragraph 36 of Archibald provides further discussion regarding DCC_{ds}. More particularly, paragraph 36 describes that the data stored in the sector body is used to generate (i) a calculated DCC_{ds} for each data sector in the sector stripe, (ii) a calculated DCC_{ps} for the parity sector in the sector stripe, and (iii) a calculated DCC_{ss} for the sector stripe based on the newly calculated DCC_{ds} and DCC_{ps}. If an inconsistency is detected between a calculated DCC_{ds} and a stored DCC_{ds} for a data sector and auto-correction for user data is enabled, the user sector body of data for the effected sector is regenerated using the remaining good data and parity sector.

The description of Archibald's invention is less than clear and a substantial argument exists as to whether Archibald is enabling and thus a prior art reference with respect to the instant application. For the purpose of this petition only, Applicants will assume that Archibald is enabling.

The instant application includes independent claims 1, 11, 21, and 22. Independent claim 1 recites in pertinent part (1) receiving a request to read data; (2) reading first parity data corresponding to first striped unit data of a first stripe in response to receiving the request; (3) generating new first parity data as a function of the first stripe unit data; (4) comparing the first parity data with the new first parity data; and (5) returning data of the first stripe unit if the first parity data compares equally to the new first parity data.

It is less than clear as to whether the calculation of DCC_{ss} or DCC_{ds} in Archibald equates to generation of new parity data as a function of the first striped unit data as recited in independent claim 1. Notwithstanding the lack of clarity, Applicants submit claim 1 is nonetheless distinguishable over Archibald. More particularly, Archibald fails to teach or fairly suggest claim 1's acts of receiving a request to read data and returning data of the first stripe unit to a computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of claim 1. Indeed it appears nothing within Archibald indicates that the process of checking the consistency of data is initiated by a request for data contained within a stripe unit.

Independent claim 11 is similar to independent claim 1. Independent claim 11 recites in pertinent part (1) reading first parity data in response to receiving a request to read data, wherein the request is received from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data. Archibald fails to teach or fairly suggest receiving a request to read data from a second computer system and returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of claim 11.

Independent claim 21 recites in pertinent part (1) reading first parity data in response to receiving a request from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data. Archibald fails to teach or fairly suggest receiving a request to read data from a second computer system and returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of claim 21.

Independent claim 22 recites in pertinent part (1) means for receiving a request to read data, wherein the request is received from a computer system; (2) means for reading first parity data in response to receiving the request; (3) means for generating new first parity data as a function of the first stripe unit data; (4)means for comparing the first parity data with the new first parity data; (5) means for returning data of the first stripe unit to the computer system if the first parity data compares equally to the new first parity data. Archibald fails to teach or fairly suggest a means for receiving a request to read data from a computer system and a means for returning data of the first stripe unit to the computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of claim 22.

Accordingly, applicants submit that independent claims 1, 11, 21, and 22 are patentably distinguishable over Archibald.

U.S. Patent 6,606,629 issued to DeKoning et al. ("DeKoning") on August 12, 2003. DeKoning relates to data structures containing sequence and revision number metadata used in mass storage data integrity-assuring technique. The sequence number of metadata identifies an input/output (IO) operation such as a full-stripe write on a redundant array of independent disks (RAID) mass storage system. The data structure may also contain revision number metadata which identifies a subsequent IO operation such as a read-modify-write on only a fractional component of the entire user data. The sequence number and revision number metadata are used in an error detection and correction technique, along with parity metadata to detect and correct silent errors arising from inadvertent data path and data drive corruption. An error to a portion of the stripe is

detected by a difference in sequence numbers for all the components of data. An error arising after an IO operation is detected by a revision number which is different from the correct revision number. The errors in both cases are corrected by using the parity metadata and the correct information from the other components of the user data.

Figure 10 of DeKoning illustrates operations that use sequence number and revision number metadata to detect and correct errors. The process shown in Figure 10 is initiated with a read request. In response, user data structures and parity data structures associated with read requests are read. In step 114, if returned volume and LBA metadata match with the read request and if the CRC and checksum are valid, then sequence numbers are checked for equivalency. Using the metadata of the sequence number and the revision number, in comparing the sequence and revision numbers of the user data structure with the sequence and revision numbers of the corresponding parity data structure at steps 118 and 120, respectively, it can be determined as a consequence of the read operation itself whether an error, data path corruption, or drive corruption has occurred.

While DeKoning describes use of his invention with respect to a RAID storage system, DeKoning does not teach or fairly claim 1's acts of (1) receiving a request to read data; (2) reading first parity data corresponding to first striped unit data of a first stripe in response to receiving the request; (3) generating new first parity data as a function of the first stripe unit data; (4) comparing the first parity data with the new first parity data; and (5) returning data of the first stripe unit if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 1.

DeKoning does not teach or fairly suggest (1) reading first parity data in response to receiving a request to read data, wherein the request is received from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 11.

DeKoning does not teach or fairly suggest (1) reading first parity data in response to receiving a request from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 21.

DeKoning does not teach or fairly suggest (1) means for receiving a request to read data, wherein the request is received from a computer system; (2) means for reading first parity data in response to receiving the request; (3) means for generating new first parity data as a function of the first stripe unit data; (4) means for comparing the first parity data with the new first parity data; (5) means for returning data of the first stripe unit to the computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 22.

Accordingly, Applicants submit that independent claims 1, 11, 22, and 22 are patentably distinguishable over DeKoning.

U.S. Patent No. 5,948,110 issued to Hitz et al. ("Hitz") on September 7, 1999. Hitz relates to a method for providing parity in a RAID subsystem using non-volatile memory. Non-volatile RAM is used to increase the speed of RAID recovery from a disk error. This is accomplished by keeping a list of all disk blocks for which the parity is possibly inconsistent. Such a list of disk blocks is much smaller than the total number of parity blocks in the RAID subsystem. Figures 11A – 11C are flow diagrams illustrating the invention of heights. In Figure 11B, a determination is made in step 1104 if there are any stripes specified in a list of dirty stripes in the RAM memory. Presuming there are, blocks are read which are needed to recompute parity as shown in step 1106. Parity for the stripe is recomputed as shown in step 1107, and the new parity is written for the stripe as shown in step 1108. In Figure 11C, all disk blocks required to update a stripe are read in block 1109. In block 1110 new parity for the stripe is calculated. The stripe number of the stripe being written to is added to the list of dirty stripes in memory in step 1111. In step 1112, disk blocks required to update the stripe are written. Finally, the stripe number of the stripe is removed from the list of dirty stripes in the memory.

While Hitz relates to providing parity in a RAID subsystem, Hitz does not teach or fairly suggest claim 1's acts of generating new first parity data as a function of the first stripe unit data, comparing the first parity data with the new first parity data, and returning data of the first stripe unit if the first parity data compares equally to the new first parity, either alone or in combination with the remaining limitations of independent claim 1.

Hitz does not teach or fairly suggest (1) reading first parity data in response to receiving a request to read data, wherein the request is received from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 11.

Hitz does not teach or fairly suggest (1) reading first parity data in response to receiving a request from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 21.

Hitz does not teach or fairly suggest (1) means for receiving a request to read data, wherein the request is received from a computer system; (2) means for reading first parity data in response to receiving the request; (3) means for generating new first parity data as a function of the first stripe unit data; (4) means for comparing the first parity data with the new first parity data; (5) means for returning data of the first stripe unit to the computer system if the first parity data compares equally to the new first parity data either alone or in combination with the remaining limitations of independent claim 22.

Accordingly, Applicants submits that independent claims 1, 11, 21, and 22 are patentably distinguishable over Hitz.

U.S. Patent No. 5,623,595 issued to Bailey ("Bailey") on April 22, 1997. Bailey relates to a method and apparatus for transparent, real time reconstruction of corrupted

data in a redundant array data storage system. In Bailey, data is read from a stripe. To read a stripe, a RAID request is sent to each storage unit in the redundancy group, requesting the block of data in that storage unit that corresponds to the stripe being read. Each storage unit independently processes its read request and sends back the requested block as soon as it has been read. Each stripe consists of N+1 blocks where one of the blocks is parity. As soon as the N blocks are received, they are XOR'd together to reconstruct the outstanding N+1 block. The N receipt blocks and the XOR to some, representing a reconstruction of the outstanding N+1 block, are delivered to the requesting device or process.

While Bailey uses a parity RAID storage subsystem, Bailey fails to teach or fairly suggest claim 1's acts of generating new first parity data as a function of the first stripe unit data, comparing the first parity data with the new first parity data, and returning data of the first stripe unit if the first parity data compares equally to the new first parity, either alone or in combination with the remaining limitations of independent claim 1.

Bailey does not teach or fairly suggest (1) reading first parity data in response to receiving a request to read data, wherein the request is received from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 11.

Bailey does not teach or fairly suggest (1) reading first parity data in response to receiving a request from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 21.

Bailey does not teach or fairly suggest (1) means for receiving a request to read data, wherein the request is received from a computer system; (2) means for reading first parity data in response to receiving the request; (3) means for generating new first parity

data as a function of the first stripe unit data; (4) means for comparing the first parity data with the new first parity data; (5) means for returning data of the first stripe unit to the computer system if the first parity data compares equally to the new first parity data either alone or in combination with the remaining limitations of independent claim 22.

Accordingly, Applicants submit that independent claims 1, 11, 21, and 22 are patentably distinguishable over Bailey.

EP Application No. 0 837 393 A2 of Gordon et al. ("Gordon") published on April 22, 1998. Gordon relates to on-line restoration of redundancy information in a redundant array system. Figure 3 is a flow chart representing the restoration process in Gordon. In block 40, an interrupted write operation for a new data block is resubmitted. The write operation is resubmitted after the write operation was interrupted. In block 41, all validated blocks in the affected stripe are read. In block 42, the new data block and the valid data blocks are XOR'd to compute the new parity block. In step 43, the new parity and new data blocks are stored in the stripe.

While Gordon relates to restoration within a RAID system, Gordon fails to teach or fairly suggest claim 1's acts of generating new first parity data as a function of the first stripe unit data, comparing the first parity data with the new first parity data, and returning data of the first stripe unit if the first parity data compares equally to the new first parity, either alone or in combination with the remaining limitations of independent claim 1.

Gordon does not teach or fairly suggest (1) reading first parity data in response to receiving a request to read data, wherein the request is received from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 11.

Gordon does not teach or fairly suggest (1) reading first parity data in response to receiving a request from a second computer system; (2) generating new first parity data as a function of first stripe unit data; (3) comparing the first parity data with the new first

parity data; (4) returning data of the first stripe unit to the second computer system if the first parity data compares equally to the new first parity data, either alone or in combination with the remaining limitations of independent claim 21.

Gordon does not teach or fairly suggest (1) means for receiving a request to read data, wherein the request is received from a computer system; (2) means for reading first parity data in response to receiving the request; (3) means for generating new first parity data as a function of the first stripe unit data; (4) means for comparing the first parity data with the new first parity data; (5) means for returning data of the first stripe unit to the computer system if the first parity data compares equally to the new first parity data either alone or in combination with the remaining limitations of independent claim 22.

Accordingly, Applicants submit that independent claims 1, 11, 22, and 22 are patentably distinguishable over Gordon.

The remaining claims of the instant application depend directly or indirectly from claims 1, 11, 22, and 22. Since claims 1, 11, 22, and 22 are patentably distinguishable over the references noted above, it follows that the dependent claims are likewise patentably distinguishable.

CONCLUSION

Applicants respectfully requests that this petition be granted, and that the present application receive expedited examination. Should any issues remain that might be subject to resolution through a telephonic interview, the Patent Office is requested to telephone the undersigned.

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop Petition, Commissioner for Patents, P. O. Box 1450, Alexandria, Virginia, 22313-1450, on 12/3/04.

Attorney for Applicant(s) — Date of Signature: 12/3/04

Respectfully submitted,

Eric A. Stephenson
Attorney for Applicant(s)
Reg. No. 38,321
Telephone: (512) 439-5093
Facsimile: (512) 439-5099